ST REPORT WEEKLY ONLINE MAGAZINE
Monday, OCT. 17, 1988
Vol II No. 57
==========

APEInc., P.O. BOX 74, Middlesex, N.J. 08846-0074

PUBLISHER
Ron Kovacs

GENERAL MANAGER
R.F.Mariano

==========

ST REPORT EDITOR: Thomas Rex Reade

PO Box 6672 Jacksonville, Florida. 32236-6672

Headquarters Bulletin Boards

ST Report North
201-343-1426

ST Report South
904-786-4176

ST Report Central
216-784-0574

ST Report West
916-962-2566

CONTENTS
========

==========

COMP-U-SERVE ~ GENIE ~ DELPHI ~ THE SOURCE

==========

From the Editor's Desk,

We are proud of Atari in one breath and scournful in another..this is what we are accused of...I do not see it this way. What I see is an accurate accounting to the reading public of what is actually happening in the ST world without an icing of sugar. Someone remarked to me, "why didn't you report the fact that Atari was at the NAMM show in Atlanta? I have but one real answer for this and it is quite simple...

ATARI HAD NOTHING TO SHOW BUT THE ECHO OF YESTERDAY'S APPLAUSE!

In fact, where Sam & Co. have been so darn busy BRAGGING about the lead the ST has in the midi fields. The ST has, in fact, LOST that lead! And....I might add, lost it miserably! We went and purchased the last eleven issues of Keyboard Magazine to see what Atari was doing to support the "Midi" aspect of the ST market. As expected, Atari was doing nothing! Thank the good sense of the third party developers for their ads exhorting the ST and their products. The gorgeous 2 page color ads for the Amiga, month after month (Atari has had 2!), and it's MIDI CAPABILITIES were very well done and indeed a thrust at the heart of Atari...as were the MAC ads and the superbly done IBM ads..and Sam (our friend?) tries to tell us to be "EVANGELISTS". HE should try it sometime!

Make no mistake about this,...The userbase, in general, in this country is tired of the behavior and attitude of Atari. They are almost arrogant in saying to heck with the US market. "WE HAVE EUROPE and are PROUD OF IT". To say what I feel in response to that attitude means nothing, YOU, the USERS, DEVELOPERS AND DEALERS have got to muster the courage and tell ATARI exactly how you feel! By saying little or nothing, you send a message to Atari saying, "all is ok, please continue on your present course". Sure, we understand that the Dealers and Developers have money involved, but even they must realize that if we don't get dead in Atari's face we all will have NOTHING! Big developers (hardware and software) are slipping away from "that bunch at ATARI" at a faster rate every month.

Here's ANOTHER Christmas Sales Season seemingly torpedoed by these..... (SNICKER).. "Giants of Industry"! Jack...send the "boys" back to school!

Sad to say, but I think Commodore was right to say "NO" to the Katzenjammer Kids!

How many of you were fortunate enough to see the magnificent performance by Sam and entourage on CIS ? I still re-read it and go into hysterics! I cannot believe Sam still tries to address the userbase for the ST as if it were ALL 10-14 year olds. My compliments to the SysO..ps of CIS once again for having used constraint in not forcing some real answers from the "PABLUM MAN" <<< BIG LAUGH >>>!!!! Also, they demonstrated real grit in remaining transparent in the face of a "Powder Puff" conference. Congratulations folks, for having maintained the REAL SysOp goal of:

Transparent but Effective management of the System.

I am angry! Nobody told you the "little" story about the FAMOUS Midi Instrument Manufacturer who approached a number of Computer Manufacturers to "private label" a production run that would be modified somewhat to allow up to eight midi output jacks, a thru jack and two input jacks. Of course this computer would still be a thorobred except that it would have exceptional midi potential. Guess what folks??? The machine is now offered for sale as an IBM compatible portable!!! The name on it is YAMAHA model C1. Sometimes, greed KILLS! The folks in Sunnyvale really need to send the "boys" home and let Daddy and the professionals save the business.

There WILL be more on this one.

Rex.....

P.S. "YAMAHA" has a nice "sound" to it. <smug grin>
I hope Atari is "smart enough" to approach the Casio folks...

****

NOTICE NOTICE NOTICE NOTICE NOTICE NOTICE NOTICE NOTICE NOTICE NOTICE

FOR A LIMITED TIME ONLY

COMPUSERVE WILL PRESENT $15.00 WORTH OF COMPLIMENTARY ONLINE TIME

to the Readers

ST REPORT ONLINE ELECTRONIC MAGAZINE

NEW USERS SIGN UP TODAY!

Call any of the St Report Official BBS numbers
(Listed at the top of ST REPORT)
or
Leave E-mail to St Report, Ron Kovacs or Rex Reade

Be sure to include your full mailing address so your
Compuserve kit can be immediately mailed to you!

Expires 11-30-88

NOTICE NOTICE NOTICE NOTICE NOTICE NOTICE NOTICE NOTICE NOTICE NOTICE
****

****
* The TORONTO ATARI FEDERATION *
* Presents *
* ----- *
* Sunday *
* THE November 6th *
* FIRST 10am to 6pm *
* CANADIAN at the [YYZ] *
* ATARI USERS Airport Hilton *
* CONVENTION 5875 Airport Rd *
* *
* Software/Hardware Vendors, Atari *
* Canada, PD Software, New Product *
* Demonstrations, Seminars, Users' *
* Groups, Developers and MORE!! *
* *
* $5.00 -Adults, $3.00 -Children. *
* *
* For more TAF information call: *
* 235-0318 [BBS] 425-5357 [msg] *
****

TANDY + ATARI ST?
==========

Connect the Tandy 3 1/2 inch drive
to the Atari ST

To connect the Tandy 3 1/2 inch external drive to the ST, the following connections must be made:

ST Floppy Port Pin	Function	Tandy Drive Pin
1	Read Data	30
2	Side 0 Select	32
3	Ground	Any Odd
4	Index	8
5	Drive 0 Select	10
6	Drive 1 Select	12
7	Ground	Any Odd
8	Motor On	16
9	Direction In	18
10	Step	20
11	Write Data	22
12	Write Gate	24
13	Track 00	26
14	Write Protect	28

These connections may be made using two methods. The easiest would be to take an ST drive cable, cut off one connector, and connect the cable directly to the 34 pin header on the drive itself. This will directly bypass the circuit board inside the Tandy drive case. The major drawback to this method is the fact that you will not be able to unplug the cable from the drive.

The other method, although more complicated, will allow you to utilize the 30 pin edge connector mounted in the back of the Tandy drive case. First remove all paths leading to the upper pads of the edge connector and all paths on the bottom side that do not lead to ground. Wire all connector pads on the bottom side together so that they all are connected to ground. At the point where the ribbon cable from the drive is attached to the circuit board, wire the necessary connections to the pads of the 30 pin edge connector. Remember, any odd number pin on the drive itself is ground, so connect one to the bottom connector pads of the circuit board being modified. Be sure to note each connector pad you are using, and it’s corresponding function. Now attach the ST drive cable to a 30 pin edge plug in accordance with the wiring accomplished on the circuit board.

Once wiring has been completed, place the jumper on the drive mechanism to the A0 position. If your computer won’t recognize the drive, try the A1 position.

The drive requires 12V 230mA and 5V 240mA for power. I use a standard PC power supply using either connectors P10 or P11.

At What PRICE? LOYALTY

by T."Rex" Reade

After having polled over 120 users of ST equipment, the conclusions were:

a)- The average investment is about three to five thousand dollars.
b)- The average age is from twenty three to fifty (big spread there).
c)- Almost all polled would gladly continue with Atari ST products.

Although this could never be termed a "professional" survey, I am sure if all were known about who was "asked" these questions most would concur that this "survey" is much more than accurate. Rest assured, we asked developers, musicians, and ordnary users . The part that is wild is..Atari could have done it's own informal survey of the ST users and found much of the same thing. Instead, the head man still treats us like kids with ice cream on our faces.

Sorry Sam, "it ain't gonna work"! The time has come for Atari to get real with the userbase.

We want:

1 - The ST products enhanced on a regular basis.
(not every "so often" or "whenever")

2 - The ST products properly promoted and distributed.
(WE are TIRED of hearing about EUROPE!)

3 - An end put to the "maypo express".
(spoon-fed pablum)

4 - True user input to the company that is evidenced in product upgrades. Like Hard Disk Partition Size!
(no more lame excuses...please!)

5 - Abandonment of the "lofty" dealer monarchy idea.
(what happened to.."Power Without the Price"?)

6 - An "in earnest" effort to promote and satisfy the U.S.A. market.
(Even if you do not have the machines to ship, you CAN talk about them and keep the interest level UP!)

In most of the ELECTRONICS INDUSTRY we find the annual release of NEW products exciting and envigorating to both the users AND the MARKET itself.

The users want to have and enjoy the very latest and the merchants need the annual revitalization in the market place as it causes a true competitive atmosphere and attractive pricing.

We all know (I think) the best part of the sales thrust for any new product is in the first year.

Atari sez:

"WE DON'T WANT TO ADVERTISE AND NOT DELIVER, IT ONLY HELPS OUR COMPETITION".

Rex...sez:

Atari is suffering from a case of the extreme "cheaps"..spend some bucks! Tell the US Marketplace you are alive and well. Advertise the software available for your machines in the ads, do something smart and support the developers in the United States of America! After all, THEY support YOU!

Common Sense and Experience show:

A)- The old adage, "OUTTA SIGHT - OUTTA MIND". Still fits...

B)- Advertisements similar to the two page spread (unusual) in Keyboard (NOV. 88) that displays the names of the software available for the ST gives the GOOD impression of presenting both the software and the hardware. [GOOD P.R.]

In this computer marketplace "HE WHO HAS THE GOLD MAKES THE RULES"!!!

Here the "gold" is Product Name RECOGNITION...nobody, in their right mind, would dispute this.

ATARI??.....AIN'T THAT THE PAC-MAN PEOPLE????
(ever heard that?....some recognition?... FOOL'S GOLD!)

Why, just 14 months ago you could walk into any decent music store and ask about Midi and whose computer handled it...you would be told Atari ST and then Mac..the Mac wasn't so popular because it was expensive and in monochrome. BUT NOW...I dare you to ask ...the answers you get will break your heart. It did mine. It makes you wonder if anybody in charge at Atari honestly gives a darn.

Here we approach the end of ANOTHER year of broken promises by the fearless (thoughtless) leaders of Atari. What do we know as far as the future of the ST?

NOTHING... ABSOLUTLY NOTHING!

Sure, we know what we have in front of us works ok, and we know what we would like to believe, but the truth is, we are all so very disappointed in the management of Atari. I am waiting and watching to "SEE" what takes place at Comdex (Fall).

While waiting, my senses are assaulted by all the fabulous advertisements for all the OTHER computers available to the US market!

It's Amazing! Why don't they have the terrible ** DRAM PROBLEMS ** and all the "Other Baloney" excuses and reasons we are being fed!

The TRUTH is..EUROPE has the TOP PRIORITY for new machine deliveries. The Lake Tahoe Conference made sure of this agreement with the European Wigs. As for the FUTURE????

Comdex will tell all folks,,,,,
meanwhile;

Hey, You at ATARI, Please... WAKE up ....Please keep OUR machines current and most of all,

PLEASE, STOP NEGLECTING US IN AMERICA!!

I AM A PROUD AMERICAN!

Rex.....

HE BELIEVES IN AMERICA!

NeXT Inc. introduces a new type of computer system aimed at higher education

NeXT Inc., of Palo Alto,

Wednesday, unveiled the NeXT Computer System, designed to meet the demanding and diverse needs of higher education.

The system encompasses the best attributes of work stations and personal computers, adds features previously found only on mainframes and introduces entirely new innovations.

"NeXT's mission is to collaborate with higher education to develop innovative, personal and affordable computer solutions for the next decade and beyond," said Steven P. Jobs, president and chief executive officer of NeXT.

"We began our product design process at key higher education centers in this country, discovering what they wanted from a computer. Based on what we heard, we have created a revolutionary learning and research environment that represents what computing will be like in the 1990s.

"Currently, there is a revolution in software development and use on college and university campuses, generating powerful concepts such as simulated environments for both research and learning. The problem is that higher education lacks a predictable computing target for software developers, which slows emergence of practical products.

"NeXT intends to provide this target by raising the lowest common denominator for standard capabilities in academic computing. In this way, we will help spur the realization of some innovative and important software ideas," Jobs said.

NeXT saw the need in higher education for a computer that combined qualities of workstations and personal computers, with cabilities far-exceeding either.

Specifically, the company took the workstation concepts of built-in networking, large standard display screens, multitasking and a robust application development environment, and designed and packaged them in a one-foot cube with personal computer-like characteristics such as affordability, efficient manufacturability and cool, quiet and reliable use.

At the same time, NeXT recognized that significant innovations were necessary to extend its computer system beyond a laundry list of impressive features.

NeXT chose to innovate in four main areas:

A mainframe on two chips:

The architectures of both workstations and personal computers contain inherent bottlenecks to higher performance that cannot be resolved by faster processors alone. To manage the flow of information within the system to yield peak efficiency, NeXT designed the ICP and OSP, two proprietary VLSI (very large-scale integration) chips that endow the system with mainframe-like capabilities.

NextStep:

Although UNIX provides powerful capabilities and is the most prevalent operating system for higher education and research, the complexity of UNIX-based computers has put them beyond the reach of almost everyone except scientists and engineers. At the same time, developing graphical application software has traditionally extracted an inordinate amount of time and expertise.

NeXT has addressed both these problems with NextStep, an object-oriented software environment. NextStep makes the power of UNIX accessible to all users, while it also significantly reduces the time, expertise and software code developers need to construct graphical, end-user applications.

Personal Optical Storage and the Digital Library: The potential for desktop computers to open the world’s knowledge to an individual has been restricted, in part, by inadequate mass storage and poor searching and indexing capabilities. To break through these restrictions, NeXT used a new storage technology called magneto-optics to create a removable, read/write/erasable 256 Megabyte Optical Disk as the Computer System’s standard mass storage device.

The Optical Disk makes possible the concept of the "Digital Library," which can comprise on-line reference and literary works, musical scores or images of photographic quality.

Included with every system is a powerful searching and indexing tool called the Digital Librarian and a "starter" Digital Library.

Sound and Music: Sound is considered a vital communication medium. As a result, NeXT has made sound capabilities integral to its computer system:

- a microphone jack for input, CD-quality stereo output.
- a powerful 10 MIPS Digital Signal Processor (DSP).
- a standard voice mail application.

To encourage the development of applications that include sound, music and voice, the system also includes the SoundKit and MusicKit.

The Sum is greater than the parts:

"Many of the NeXT Computer System's individual components represent major technological breakthroughs," Jobs said.

"Taken in sum, they generate capabilities and potential exceeding that of any existing category of computer system."

The system's basic hardware configuration includes the computer, a one-foot cube that houses on a single board all the computer's highly integrated silicon chips; the 256 Megabyate Optical Disk for editable storage and retrieval of vast amounts of information; the 17-inch, extremely high-resolution MegaPixel Display; and the 400 dpi Laser Printer, which is the first affordable PostScript laser printer and the first low-cost laser printer to provide 400 dots per inch (dpi) resolution.

Underlying all the system's capabilities is a small, powerful and efficient set of computer chips, all of which are standard and fit onto a single board. There are three high-performance processors in every system.

The main processor is Motorola's top-of-the-line microprocessor, the 68030.

Accompanying it is Motorola's 68882 Floating-Point Unit, for fast mathematical computations. Both these chips run at 25 megahertz.

The third processor is a 10 MIPS Motorola 56001 Digital Signal Processor chip, for real-time sound and array processing. The board can also support up to 16 megabytes (MB) of main memory.

Two proprietary VLSI chips, designed by NeXT, give the system its mainframe-like qualities.

The Integrated Channel Processor (ICP) manages the flow of data among the central processing unit (the 68030), main memory and all peripheral devices. By offloading the 68030 and ensuring the efficient flow of data within the system, the ICP allows the 68030 to run at its full rated capacity of 5 MIPS.

The ICP provides 12 dedicated DMA (direct memory access) channels, including channels for Ethernet networking and for disks, monitor, printer and other peripheral devices. The single ICP chip replaces several hundred chips performing similar functions on a mainframe computer, and it raises sustained system throughput to a level impossible with either personal computer or workstation architectures.

The other VLSI chip, the Optical Storage Processor, controls the 256 Megabyte Optical Disk, making possible this new storage technology.

The Optical Disk combines the vast storage capacities, removability and reliability of laser technology with the fast access and full read/write/erase capabilities of Winchester (magnetic) technology.

The Optical Disk provides unprecedented information storage, manipulation and retrieval. With the Optical Disk working in conjunction with the Digital Librarian, a specially designed searching and indexing tool, users can almost instantaneously locate any textual information, in any form, anywhere in the computer.

They can also browse through the system to uncover information,

ideas or connections between concepts. Software as Part of the System

NeXT includes an unparalled amount of software in the price of every NeXT Computer System. The software starts with Mach, an advanced multitasking operating system compatible with 4.3BSD UNIX, which is the standard operating system in higher education communities.

In addition, the NeXT Computer System includes NextStep, a complete software environment consisting of four components: the Window Server, the Workspace Manager, the Application Kit and the Interface Builder.

The object-oriented environment was developed with the Objective-C programming language, from the Stepstone Corp.

NextStep solves the two major problems with UNIX-based systems:

They are too complex and difficult for most non-programmers to use. They require developers to spend an inordinate amount of time and expertise creating graphical, end-user applications.

For users, NextStep makes the power of UNIX available by substituting a window-based, graphical and intuitive interface for the traditional UNIX comand-line interface. For developers, NextStep includes the Application Kit, a set of interacting software "objects" for building applications.

Also included in NextStep is Interface Builder, a completely new kind of software development tool. Interface Builder works graphically, letting the developer construct an application by choosing from a palette of available objects and using the mouse and keyboard to modify the objects as needed, define the layout and establish connections between objects.

This process permits the rapid construction of graphical user interfaces and makes application development accessible to a much larger community.

NextStep uses the Display PostScript system to ensure true WYSIWYG (What You See Is What You Get) between the screen and the printer.

The Display PostScript system includes a high-performance implementation of the PostScript language, the de-facto imaging standard for printing. It simplifies the programming of graphical applications that support high-quality printing.

To further aid developers, the NeXT Computer System includes the SoundKit, MusicKit, array processing routines, assemblers, compilers, debuggers and a terminal emulator.

Standard with each system, on the 256 Megabyte Optical Disk, is a basic Digital Library. A Digital Library can contain complete reference works, books, images or musical scores.

The bundled library includes the Webster’s Ninth New Collegiate Dictionary, including definitions, pronunciations and illustrations, not just spelling; Webster’s Collegiate Thesaurus; the Oxford Dictionary of Quotations; the Oxford University Press edition of William Shakespeare; The Complete Works.

NeXT technical references and other pertinent technical references.

The NeXT Computer System also includes a rich set of bundled application software. These applications include WriteNow, a full-featured word processing program; Mathematica, a symbolic mathematics program; the powerful NeXT SQL Database Server, from Sybase; Allegro CL Common Lisp; Jot, a personal text database manager; and a graphical electronic mail application with integrated voice mail capabilities.

NeXT has built its business plan and products to meet the needs of higher education. The company determined these needs through close collaboration with leaders at college and university campuses nationwide, uncovering the gaps between current and ideal computer technology for this marketplace.

"Higher education is a huge market, certainly big enough in itself to grow NeXT to critical mass," said Dan Lewin, vice president of sales and marketing and NeXT. "Beyond that, higher education is the most demanding and diverse marketplace conceivable. It provides a real acid test. If we can do well here, were can do well anywhere.

"The key is understanding and committing to a business model that works the way higher education does, both in its generic form and as it varies from campus to campus. That's where NeXT has the edge, because we are the only computer company that has amassed both the market knowledge and the technological ability to deliver the right computing tools," Lewin said.

During 1988, NeXT will market its computer System directly to several dozen of the nation's top institutions and software developers. NeXT expects to appeal to higher education on the strength of its technology tools and through the personal business relationships the company has established with the higher education community.

Price and Availability

The standard NeXT Computer System configuration, which includes 8 MB of main memory, the 256 Megabyte Optical Disk, the MegaPixel Display, keyboard, mouse and complete system software, will sell for $6,500.

The 400 dpi Laser Printer will sell for $2,000.

All prices quoted are for higher education.

NeXT will ship systems to its key customers and developers starting this quarter, and expects to ship systems with final software by the second quarter of 1989 to a broader base of institutions and developers.

Available options to the standard configuration include 4 MB RAM expansion modules (up to 16 MB total), 660 MB and 330 MB high-performance Winchester drives, an Ethernet kit, blank Optical Disks and printer toner cartridges.

NeXT Inc., of Palo Alto, was founded in October 1985 by Steven P. Jobs, co-founder and former chairman of Apple Computer Inc., and five other individuals. The mission of the privately held company is to collaborate with higher education to develop innovative, personal and affordable computer solutions for the 1990s and beyond.

Ed. Note:
How sweet it is to see a Computer Company have some REAL FAITH in the United States of America. Can IBM, Apple, Commodore and ALL the companies

be so totally wrong and ATARI be so right??? I think NOT! Atari had better get with the program, or be left in the "micro-dust"!

UPBEAT MIDI-BEAT

A look at a superb Midi-Utility

OMNIBANKER ST

OMNI-BANKER ST is a "universal" MIDI Librarian that is capable of loading and storing data from almost any midi device. It uses a custom instrument database (that handles several dozen different devices) to automatcally request and transmit data. It goes beyond generic librarians, by allowing you to view two banks of 32 named patches at the same time (or to zoom in and view 1 bank of 128 patches) and arrange patches both within and between banks. Also it is mouse or keyboard driven and it permits selection of individual patches for audition or editing.

When you open the instrument area of the program, you will immmediately know you are in GOOD company... I saw all the favorites there!

ENSONIQ	YAMAHA	CASIO,	KORG
AKAI	ART	J.L.COOPER	KAWAI
OBERHEIM	ROLAND	SEQUENTIAL	SEIL
			VOYETRA

More added quarterly and, UPDATES ARE FREE!.

Omnibanker is written especially for the ST and is fully GEM based. The fine design of the program allows you to use eirther a program version or a desktop accessory version. Incidently, the Acc. version has ALL the functions of the program.

The normal retail price of OMNI-BANKER ST is $95.00 plus 5.00 for shipping and handling....(10.00 Canada) ..Omnibanker will ber appearing on the dealer shelves in December.

:SPECIAL OFFER FOR THE READERS OF ST REPORT:

PARADGM SOFTWARE PRODUCTS have authorized us at ST Report to make this extraordinary offer to our reader:

Until 12/01/88 OMNIBANKER ST WILL BE OFFERED TO OUR READERS FOR:

*** $55.00 + S&H! ***

CALL: 616-372-5972
Paradigm Software Products
1369 Concord Place, Suite 3-B
Kalamazoo, Michigan 49009

Please ask for Mark , and tell him you saw this mentioned in ST Report.

Oh! yes..I almost forgot, there is a 30 day money back guarantee!

ST REPORT CONFIDENTIAL
==========

Sunnyvale, CA ----------	There is some very STRONG talk about a slick move to relocate corporate hdqtrs, something about "more than one set of offices".
Buena Park, CA ----------	Yamaha, a leader in Midi Instrument Manufacture, has a FREE newsletter available send to: AFTERTOUCH, PO BOX 7938, Northridge,CA 91327-7938.
Hollywood, CA ----------	Paramount Studios hosts this year's MAC-MUSIC FEST 2.0 Co-Sponsored by Filmsonix, Dec 3-4.
Los Angeles, CA ----------	Hybrid Arts, a staunch supporter of the Atari Midi line has invested a small fortune in music industry ads..Call their BBS: 213-826-4288.
Menlo Park, CA ----------	Digidesign and C-Lab Software are also investing in strong holiday ads for their midi products and the ST. Check the latest in Keyboard Magazine.
Buena Park, CA ----------	The YAMAHA/TOSHIBA 80286 C1 MIDI computer (portable) has all the hot features. Reportedly, Yamaha approached the portable market with an open mind on which way to go, IBM is their choice. Dont look now folks, but guess who will lead the "PACK" in midi.
Sunnyvale, CA ----------	Is the Atari Laptop (portable) really their own? Folks, when you see the new Lap Top, look close and you tell us.
Houston, TX ----------	The C of C here is thoroghly worn out trying and will now adopt a blase attitude toward Atari's games they call negotiations. In other words, STILL, NO INK ON THE PAPER!
NEW YORK, NY ----------	Acme Toys is releasing a new pull toy, it pulls a wagon filled with computers, wears a beanie with a propeller on it, it's bird! It sings; "cheep, cheep!" And it's T-Shirt sez "we are for the birds"....

Sounds familiar to me....how about you?

FILE TYPES AND FILENAME EXTENSIONS

What needs what to run?

By Robert J. Retelle
(c)1987

There are many different types of files available for the Atari ST, from On-Line Services, from local Computer Bulletin Boards, and from User Group Software Libraries, some of which are ready-to-run programs, but many of which require some kind of preparation before they can be used.

Generally the Description of the software will tell you if the file or program needs any special treatment, or if any other files are required.

The most important indicator of what the file is though, is the 'Filename Extension'

The Atari ST computer allows disk files to have NAMES consisting of up to eight letters or numbers, and an EXTENSION consisting of three more letters or numbers. The NAME is separated from the EXTENSION by a period:

FILENAME.EXT MEGAROID.PRG

You can't use spaces or most other characters in the filenames. You CAN use the underline character to separate words if you wish:

GAME_1.PRG HI_SCOR.DAT

Usually, the NAME will tell you what the file is, and the EXTENSION will tell you what TYPE of file it is.

There are three types of files which can be RUN on the Atari ST computer directly:

.PRG .TOS .TTP

.PRG Files are generally PROGRAMs which use GEM (windows, menus and the mouse). You run these programs by double clicking on their ICON or FILENAME from an open directory window.

.TOS
Files are generally PROGRAMS which do NOT use the GEM desktop. You run them the same way as .prg files.

.TTP
Files are special versions of TOS programs which need certain input from you before they can run (TTP means TOS Takes Parameters).

When you run these programs by clicking on them, a dialogue box will open on the screen and you can type the needed information on a line in the box. You will need to know, either from the File Description in the Library, or from reading the Documentation for the program, what to type for a particular program.

Note that while these programs will RUN on the ST, they may need other files on the same disk with them to work properly. The File Description or Documentation should tell you if other files are required.

Files which might be required by a PROGRAM include:

.RSC .DAT .PIC

.RSC

Files are RESOURCE files used by GEM programs, and contain information needed by the program to run. The .RSC file MUST be on the same disk as the .PRG file, and must have the same FILENAME as the .PRG file:

MEGAROID.PRG MEGAROID.RSC

.DAT

Files are DATA files, to be used by other programs. You might see these called .D8A files occasionally. They generally must be on the same disk as the program which will use them.

.PIC

Files are PICTURE files which are loaded onto the screen by the program being run. Again, these generally must be on the same disk as the program.

There may be other types of files required to run a program, but these are the most common ones. If you find a file with an unusual EXTENSION, it's possible that it is one of these required files.

There is a special form of program for the ST, called ACCessories. These are programs which are loaded into the computer's memory when the computer is first turned on, stay there, ready to be used from within other GEM based programs. ACCessories may include calculators, clocks, notepads and other useful applications. They allow you to switch to other functions without leaving the program you are running. There are even Game Accessories which allow you to play games while you're supposed to be working with Spreadsheets or Word Processors...! To load an accessory, copy it onto the disk you use when you first turn on your computer. Up to six acc files can be on this disk, allowing up to six choices of acc in your programs. Note that Acc take up memory in your computer, which will decrease the amount of memory available to other programs. Accessories are identified by the Filename Extension: .ACC

Another common type of file is one which contains TEXT or DOCUMENTATION in a readable form:

.TXT .DOC READ.ME

Generally these will be information or instructions needed to run a program. These files can usually be read directly from the desktop by clicking on their icon or filename, and selecting SHOW from the alert box which pops up on the screen. You can also usually print these files on

your printer by selecting PRINT from the same alert box.

The READ.ME file is usually included on a program disk or as part of a group of files to give you important information about new features or updates. You *should* read it before trying to use the program!

Note that the files created by WORD PROCESSORS may NOT be readable from the desktop SHOW function because of the special formats they save the text with.
One example is files identified with a .STW entension. These are created with the STWRITER program, and need to be loaded back into that word processor to be read or printed.

Also note that files created with the Publishing Partner desktop publishing program have a .DOC (Document) extension, but are NOT readable from the SHOW option of GEM.

At times, programs may come with files containing the original 'source code' that was used to write the program. These files are usually not necessary to just run the program, and are only of interest if you would like to see how the program was written, or if you would like to modify the program.

These SOURCE CODE or programming files are generally identified:

.C .MOD .PAS .ASM .H .SRC

Programs written in certain languages, such as BASIC or LOGO cannot be run by themselves. They need to have the LANGUAGE itself loaded into the computer and run first, then the actual program can be loaded and run by the language.

These programs are identified:

.BAS .LOG

Note that there are now several different BASIC languages in use on the ST, and the programs for each of them are generally not interchangable with other kinds of BASIC. The description of the file in the Software Library should tell you which BASIC language you will need to run the program.

MUSIC AND GRAPHICS FILES

There are several kinds of files for various Music and Graphics programs available in software libraries. In most cases, the filename extension will help identify the kind of program you will need to use to display the picture, or play the music.

.SNG
   Currently, the most popular music program seems to be The Music Studio from Activision. The great majority of music files available require the use of the Music Studio program to play. There is as yet NO Public Domain player program for these song files. You must first load and run your Music Studio program, then load and play the .SNG files.

.MCS
   MUSC CONSTRUCTION SET Song files.

.NEO
This identifies a file as a picture drawn using Atari's NEOchrome drawing program. These files are always in LOW resolution, and may contain 'color rotation' animation effects. NEOchrome files are always contain 32128 bytes. NEOchrome pictures can be displayed by loading them into the NEOchrome program, or by using one of several picture display programs in the Software Library. One good choice is EFFETS.PRG.

.PI1
This is a LOW resolution picture drawn using the DEGAS drawing program. DEGAS pictures are always 32034 bytes long. You can use DEGAS itself to display the pictures, or a 'stand alone' display program. DEGASPIC.PRG is a good one to use.

.PI2 This is a MEDIUM resolution DEGAS picture

.PI3 This is a HIGH resolution DEGAS picture.

.PC1 .PC2 .PC3
These are drawings done with the DEGAS ELITE drawing program, and saved in COMPRESSED format. They corespond to the same resolutions as the DEGAS files with the same number in the filename extension. Because they are compressed, they will be smaller files than the normal DEGAS files. The actual number of bytes will depend on the complexity of the picture. Both compressed and uncompressed DEGAS ELITE pictures can be displayed by using SHOWPIC2.PRG.

.TNY
This indicates a picture which has been compressed using Dave Mumper's TINYSTUFF program, and is referred to as TINY format. These pictures can be either NEOchrome or DEGAS originally. If the original picture was a NEOchrome drawing with color rotation animation, the animation effects will be preserved in the .TNY file. To display TINY format pictures, use TINYVIEW.PRG.

.TN1 .TN2 .TN3
These are picture files compressed with Dave Mumper's new TINYSTUFF2 system. Use TNYVIEW2.PRG to view them.

COMPRESSED FILE FORMATS

ARChive, SQUeeze and LIBrary

To save space, and the time it takes to download files, many of the files and programs in software ibraries have been 'compressed' using special techniques. Many times, this will result in a file that is up to half size, meaning you can download it in half the time!

Such a file is useless to you however, until you UNcompress it in your own computer after downloading it. This extra step is more than justified by the savings of time and money resulting from the smaller file size.

There are three common methods of compression that you will encounter in the Software Libraries. Files processed with each will be identified by the Filename Extender:

.ARC .xQx .LBR

.ARC
This indicates that the file has been processed with the ARChive utility. ARChive has been adopted as the standard compression method in most software libraries, replacing the other two on most new files. ARChive not only allows files to be compressed, but allows several files to be joined into one smaller, easier to handle package. For example, a program may require a .PRG program, a .DAT data file, a .PIC picture and have a .DOC documentation file. Using ARChive, all four files can be compressed into one .ARC file, making it easy to download the entire group of files at once. Once you have downloaded the .ARC file, you must use ARC.TTP to 'unARChive' it. Another file you may need to make the best use of the ARChive system is ARC.ARC which included the instructions for using ARC.TTP

SUMMARY OF FILETYPES

.PRG Runnable GEM program
.TOS Runnable TOS program
.TTP Runnable program which requires input

.RSC Resource file required to be on the disk with its .PRG program
.DAT Data file used by a program
.PIC Picture file used by a program

.TXT Readable text file
.DOC Readable Documentation (instructions) file

.C
.MOD
.PAS
.ASM
.SRC Source Code files for programmers

.BAS BASIC program. Requires a BASIC Language to be loaded first
.LOG LOGO program. Requires LOGO language to be loades first

.SNG Music Studio music file. Requires Activision Music Studio program

.NEO NEOchrome drawing. Use EFFETS.PRG to view

.PI1
.PI2
.PI3 Degas drawing. Use DEGASPIC.PRG to view

.PC1
.PC2
.PC3 Compressed DEGAS Elite drawing. Use SHOPIC2.PRG to view

.TNY Compressed TINY format picture. Use TINYVIEW.PRG to view

.TN1
.TN2

.TN3 Compressed TINY2 format picture. Use TNYVIEW2.PRG to view

.ARC ARChived file. Use ARC.TTP to restore.

Professional GEM by Tim Oren
Column #8 - User Interfaces, Homily #1

AND NOW FOR SOMETHING COMPLETELY DIFFERENT!

In response to a number of requests, this installment of ST PRO GEM will be devoted to examining a few of the principles of computer/human interface design, or "religion" as some would have it. I'm going to start with basic ergonomic laws, and try to draw some conclusions which are fairly specific to designing for the ST. If this article meets with general approval, further "homilies" may appear at irregular intervals as part of the ST PRO GEM series.

For those who did NOT ask for this topic, it seems fair to explain why your diet of hard-core technical information has been interrupted by a sermon! As a motivater, we might consider why some programs are said by reviewers to have a "hot" feel (and hence sell well!) while others are "confusing" or "boring".

Alan Kay has said that "user interface is theatre". I think we may be able to take it further, and suggest that a successful program works a bit of magic, persuading the user to suspend his disbelief and enter an imaginary world behind the screen, whether it is the mathematical world of a spreadsheet, or the land of Pacman pursued by ghosts.

A reader of a novel or science fiction story also suspends disbelief to participate in the work. Bad grammar and clumsy plotting by the author are jarring, and break down the illusion. Similarly, a programmer who fails to pay attention to making his interface fast and consistent will annoy the user, and distract him from whatever care has been lavished on the functional core of the program.

CREDIT WHERE IT'S DUE

Before launching into the discussion of user interface, I should mention that the general treatment and many of the specific research results are drawn from Card, Newell, and Moran's landmark book on the topic, which is cited at the end of the article. Any errors in interpretation and application to GEM and the ST are entirely my own, however.

FINGERTIPS

We'll start right at the user's fingers with the basic equation governing positioning of the mouse, Fitt's Law, which is given as

```
T = I * LOG2( D / S + .5)
```

where T is the amount of time to move to a target, D is the distance of the target from the current position, and S is the size of the target, stated in equivalent units. LOG2 is the base 2 (binary) logarithm function, and I is a proportionality constant, about 100 milliseconds per bit, which corresponds to the human's "clock rate" for making incremental movements.

We can squeeze an amazing amount of information out of this formula when attempting to speed up an interface. Since motion time goes up with distance, we should arrange the screen with the usual working area near the center, so the mouse will have to move a smaller distance on average from a selected object to a menu or panel. Likewise, any items which are usually used together should be placed together.

The most common operations will have the greater impact on speed, so they should be closest to the working area and perhaps larger than other icons or menu entries. If you want to have all other operations take about the same time, then the targets farthest from the working area should be larger, and those closer may be proportionately smaller.

Consider also the implications for dialogs. Small check boxes are out. Large buttons which are easy to hit are in. There should be ample space between selectable items to allow for positioning error. Dangerous options should be widely separated from common selections.

MUSCLES

Anyone who has used the ST Desktop for any period of time has probably noticed that his fingers now know where to find the File menu. This phenomenon is sometimes called "muscle memory", and its rate of onset is given by the Power Law of Practice:

```
T(n) = T(1) * n ** (-a)
```

where T(n) is the time on the nth trial, T(1) is the time on the first trial, and a is approximately 0.4. (I have appropriated ** from Fortran as an exponentiation operator, since C lacks one.)

This first thing to note about the Power Law is that it only works if a target stays in the same place! This should be a potent argument against rearranging icons, menus, or dialogs without some explicit request by the user. The time to hit a target which moves around arbitrarily will always be T(1)!

In many cases, the Power Law will also work for sequences of operations to even greater effect. If you are a touch typist, you can observe this effect by comparing how fast you can enter "the" in comparison to three random letters. We'll come back shortly to

consider what we can do to encourage this phenomenon.

EYES

Just as fingers are the way the user sends data to the computer, so the eyes are his channel from the machine. The rate at which information may be passed to the user is determined by the "cycle time" of his visual processor. Experimental results show that this time ranges between 50 and 200 milliseconds.

Events separated by 50 milliseconds or less are always perceived as a single event. Those separated by more than 200 milliseconds are always seen as separate. We can use these facts in optimizing user of the computer's power when driving the interface.

Suppose your application's interface contains an icon which should be inverted when the mouse passes over it. We now know that flipping it within one twentieth of a second is necessary and sufficient. Therefore, if a "first cut" at the program achieves this performance, there is no need for further optimization, unless you want to interleave other operations. If it falls short, it will be necessary to do some assembly coding to achieve a smooth feel.

On the other hand, two actions which you want to appear distinct or convey two different pieces of information must be separated by an absolute minimum of a fifth of a second, even assuming that they occur in an identical location on which the user's attention is already focused.

We are able to influence the visual processing rate within the 50 to 200 millisecond range by changing the intensity of the stimulus presented. This can be done with color, by flashing a target, or by more subtle enhancements such as bold face type. For instance, most people using GEM soon become accustomed to the "paper white" background of most windows and dialogs. A dialog which uses a reverse color scheme, white letters on black, is visually shocking in its starkness, and will immediately draw the user's eyes.

It should be quickly added that stimulus enhancement will only work when it unambiguously draws attention to the target. Three or four blinking objects scattered around the screen are confusing, and worse than no enhancement at all!

SHORT-TERM MEMORY

Both the information gathered by the eyes and movement commands on their way to the hand pass through short-term memory (also called working memory). The amount of information which can be held in short-term memory at any one time is limited. You can demonstrate this limit on yourself by attempting to type a sheet of random numbers by looking back and forth from the numbers to the screen. If you are like most people, you will be able to remember between five and nine numbers at a time. So universal is this finding that it is sometimes called "the magic number seven, plus or minus two".

This short-term capacity sets a limit on the number of choices which the user can be expected to grasp at once. It suggests that the number of independent choices in a menu, for instance, should be

around seven, and never exceed nine. If this limit is violated, then the user will have to take several glances, with pauses to think, in order to make a choice.

CHUNKING

The effective capacity of short-term memory can be increased when several related items are mentally grouped as a "chunk". Humans automatically adopt this strategy to save themselves time. For instance, random numbers had to be used instead of text in the example above, because people do not type their native language as individual characters. Instead, they combine the letters into words and remember these chunks instead. Put another way, the characters are no longer considered as individual choices.

A well designed interface should promote the use of chunking as a strategy by the user. One easy way is to gather together related options in a single place. This is one reason that like commands are grouped into a single menu which is hidden except for its title. If all of the menu options were "in the open", the user would be overwhelmed with dozens of alternatives at once. Instead, a "Show Info" command, for instance, becomes two chunks: pick File menu, then pick Show.

Sometimes the interface can accomplish the chunking for the user. Consider the difference between a slider bar in a GEM program, and a three digit entry field in a text mode application. Obviously, the GEM user has fewer decisions to make in order to set the associated variable.

THINK!

While we are puttering around trying to speed up the keyboard, the mouse, and the screen, the user is actually trying to get some work done. We need to back off now, and look at the ways of thinking, or cognitive processes, that go into accomplishing the job.

The user’s goal may be to enter and edit a letter, to retrieve information from a database, or simply draw a picture, but it probably has very little to do with programming. In fact, the Problem Space Principle says that the task can be described as a set of states of knowledge, a set of operators and associated constraints for changing the states, and the knowledge to choose the appropriate operator, which resides in the user’s head.

Those with a background in systems theory can consider this as a somewhat abstract, but straightforward, statement in terms of state variables and operators. A programmer might compare the knowledge states to the values of variables, the operators to arithmetic and logic operations, the constraints to the rules of syntax, and the user’s knowledge to the algorithm embodied by a program.

ARE WE NOT MEN?

A rational person will try to attain his goals (get the job done) by changing the state of his problem space from its initial state to the goal state. The initial state, for instance, might be a blank

word processor screen. The desired final state is to have a completed business letter on the screen.

The Rationality Principle says that the user's behavior in typing, mousing, and so on, can be explained by considering the tasks required to achieve the goal, the operators available to carry out the tasks, and the limitations on the user's knowledge, observations, and processing capacity. This sounds like the typical user of a computer program must spend a good deal of time scratching his head and wondering what to do next. In fact, one of Card and Moran's key results is that this is NOT what takes place.

What happens, in fact, is that the trained user strikes a sort of "modus vivendi" with his tool and adopts a set of repetitive, trained behavior patterns as the best way to get the job done. He may go so far as to ignore some functions of the program in order to set up a reliable pattern. What we are looking for is a way of measuring and predicting the "quality" of this trained behavior. Since using computers is a human endeavor, we should consider not only the speed with which the task is completed, but the degree of annoyance or pleasure associated with the process.

Card and Moran constructed a series of behavioral models which they called GOMS models, for Goals-Operators-Methods-Selection. These models suggested that in the training process the user learned to combine the basic operators in sequences (chunks!) which then became methods for reaching the goals. Then these first level methods might be combined again into second level methods, and so forth, as the learning progressed.

The GOMS models were tested in a lengthy series of trials at Xerox PARC using a variety of word processing software. (Among the subjects of these experiments were the inventors of the windowing methods used in GEM!) The results were again surprising: the level of detail in the models was really unimportant!

It turned out to be sufficient to merely count up the number of keystrokes, mouse movements, and thought intervals required by each task. After summing up all of the tasks, any extra time for the computer to respond, or the user to move his hands from keyboard to mouse, or eyes from screen to printed page is added in. This simplified version is called the Keystroke-Level Model.

As an example of the Keystroke Model, consider the task of changing a mistyped letter on the screen of a GEM word processor. This might be broken down as follows: 1) find the letter on the screen; 2) move hand to mouse; 3) point to letter; 4) click mouse button; 5) move hand to keyboard; 6) strike "Delete" key; 7) strike key for new character.

The sufficiency of the Keystroke Model is great news for our attempt to design faster interfaces. It says we can concentrate our efforts on minimizing the number of total actions to be taken, and making sure that each action is as fast as possible. We have already discussed some ways to speed up the mouse and keyboard actions, so let's now consider how to speed up the thought intervals, and cut the number of actions.

One way to cut down "think time" is to make sure that the capacity of short-term memory is not exceeded during the course of a task. For example, the fix-a-letter task described above required the user to

remember 1) his place in the overall job of typing the document; 2) the task he is about to perform; 3) where the bad character appeared, and 4) what the new character was. When this total of items creeps toward seven, the user often loses his place and commits errors.

You can appreciate the ubiquity of this problem by considering how many times you have made mistakes nesting parentheses, or had to go back to count them, because too many things happened while typing the line to remember the nesting levels. The moral is that operations with long strings of operands should be avoided when designing an interface.

The single most important factor in making an interface comfortable to use is increasing its predictability, and decreasing the amount of indecision present at each step during a task. There is (inevitably) an Uncertainty Principle which relates the number of choices at each step to the associated
time for thought:

T = I * LOG2 (N + 1)

where LOG2 is the binary logarithm function, N is the number of equally probable choices, and I is a constant of approximately 140 msec/bit. When the alternates are not equally probable, the function is more complex:

T = I * SUM-FOR-i-FROM-1-TO-N (P(i) * LOG2(1 / P(i) + 1))

where the P(i) are the probabilities of each of the choices (which must sum to one). (SUM-FOR-i... is the best I can do for a sigma operator on-line!) Those of you with some information theory background will recognize this formula as the entropy of the decision; we'll come back to that later.

So what can we learn from this hash? It turns out, as we might expect, that we can decrease the decision time by making some of the user's choices more probable than others. We do that by means of feedback cues from the interface.

The important of reliable, continuous meaningful feedback cannot be emphasized enough. It helps the beginner learn the system, and its predictability makes the program comfortable for the expert. Programs with no feedback, or unreliable cues, produce confusion, dissonance, and frustration in the user.

This principle is so important that I going to give several examples from common GEM practice. The Desktop provides several instances. When an object is selected and a menu drops down, only those choices which are legal for the object are in black. The others are dimmed to grey, and are therefore removed from the decision. When a pick is made from the menu, the bar entry remains black until the operation is complete, reassuring the user that the correct choice was made. In both the Desktop and the RCS, items which are double-clicked open up with a "zoom box" from the object, again showing that the right object was picked.

Other techniques are useful when operator icons are exposed on the screen. When an object is picked, the legal operations might be outlined, or the bad choices might be dimmed. If the screen flashing produced by this is objectionable, the legal icons can be made mouse

sensitive, so they will "light up" when the cursor passes over - again showing the user which choices are legal.

The desire for feedback is so strong that it should be provided even while the computer is doing an operation on its own. The hour glass mouse form is a primitive example of this. More sophisticated are "progress indicators" such as animated thermometer bars, clocks, or text displays of the processing steps. The ST Desktop provides examples in the Format and Disk Copy functions. The purpose of all of these is to reassure the user that the operation is progressing normally. Their lack can lead to amusing spectacles such as secretaries leaning over to hear if their disk drives are working!

Another commonly overlooked feature is error prevention and correction. Card and Moran's results showed that in order to go faster, people will tolerate error rates of up to 30% in their work. Any program which does not give a fast way to fix mistakes will be frustrating indeed!

The best way to cope with an error is to "make it didn't happen", to quote a common child's phrase. The same feedback methods discussed above are also effective in preventing the user from picking inappropriate combinations of objects and operations. Replacement of numeric type-ins with sliders or other visual controls eliminates the common "Range Error". The use of radio buttons prevents the user from picking incompatible options. When such techniques are used consistently, the beginner also gains confidence that he may explore the program without blundering into errors.

Once an error has occured, the best solution is to have an "inverse operation" immediately available. For instance, the way to fix a bad character is to hit the backspace key. If a line is inadvertantly deleted, there should be a way to restore it.

Sometimes the mechanics of providing true inverses are impractical, or end up cluttering the interface themselves. In these cases, a global "Undo" command should be provided to reverse the effect of the last operation, no matter what it was.

OF MODES AND BANDWIDTH

Now I am going to depart from the Card, Newell and Moran thread of discussion to consider how we can minimize the number of operations in a task by altering the modes of the interface. Although "no modes" has been a watchword of Macintosh developers, the term may need definition for Atarians.

Simply stated, a mode exists any time you cannot get to all of the capabilities of the program without taking some intermediate step. Familiar examples are old-style "menu-driven" programs, in which user must make selections from a number of nested menus in order to perform any operation. The options of any one menu are unavailable from the others.

Recall that the user is trying to accomplish work in his own problem space, by altering its states. A mode in the program adds additional states to the problem space, which he is forced to consider in order to get the job done. We might call an interface which is completely modeless "transparent", because it adds no states between

the user and his work. One of the best examples of a transparent program is the 15-puzzle in the Macintosh desk accessory set. The problem space of rearranging the tiles is identical between the program and a physical puzzle.

Unfortunately, most programmers find themselves forced to put modes of some sort into their programs. These often arise due to technological limitations, such as memory space, screen "real estate", or performance limitations of peripherals. The question is how the modes can be made least offensive.

I will make the general claim that the frustration which a mode produces is directly proportional to the amount of the user's bandwidth which it consumes. In other words, we need to consider how many keystrokes, mouse clicks, eye movements, and so on, are going into manipulating the true problem states, and how many are being absorbed by the modes of the program. If the interface is wasting a large amount of the user's effort, it will be perceived as slow and annoying.

Here we can consider again the hierarchy of goals and methods which the user employs. When the mode is low in the hierarchy, and close to the user's "fingertips", it is encountered the most frequently. For instance, consider how frustrating it would be to have to hit a function key before typing in each character!

The "menu-driven" style of programs mentioned above are almost as bad, since usually only one piece of information is collected at each menu. Such a program becomes a labyrinth of states better suited to an adventure game!

The least offensive modes are found at the higher, goal related levels of the hierarchy. The better they align with changes in the state of the original problem, the more they are tolerated. For example, a word processing program might have one screen layout for program editing, another for writing letters, and yet another while printing the documents. A multi-function business package might have one set of menus for the spreadsheet, another for a graphing module, and a third for a database.

In some cases the problem solved by the program has convenient "fracture lines" which can be used to define the modes. An example in my own past is the RCS, where the editing of each type of resource tree forms its own mode, with each of the modes nested within the overall mode and problem of composing the entire resource tree.

TO DO IS TO BE!

Any narrative description of user interface is bound to be lacking. There is no way text can convey the vibrancy and tactile pleasure of a good interface, or the sullen boredom of a bad one. Therefore, I encourage you to experiment. Get out your favorite arcade game and see if you can spot some of the elements I have described. Dig into your slush pile for the most annoying program you have ever seen, run it and see if you can see mistakes. How would you fix them? Then... go do it to your own program!

AMEN...

This concludes the sermon. I'd like some Feedback as to whether you found this Boring Beyond Belief or Really Hot Stuff. If enough people are interested, homily number two will appear a few episodes from now. The very next installment of ST PRO GEM will go back to basics to explore VDI drawing primitives. In the meantime, you might investigate some of the Good Books on interface design referenced below.

REFERENCES

Stuart K. Card, Thomas P. Moran, and Allen Newell, THE PSYCHOLOGY OF HUMAN-COMPUTER INTERACTION, Lawrence Erlbaum Associates, Hillsdale, New Jersey, 1983. (Fundamental and indispensible. The volume of experimental results make it weighty. The Good Parts are at the beginning and end.)

"Macintosh User Interface Guidelines", in INSIDE MACINTOSH, Apple Computer, Inc., 1984. (Yes, Atarians, we have something to learn here. Though not everything "translates", this is a fine piece of principled design work. Read and appreciate.)

James D. Foley, Victor L. Wallace, and Peggy Chan, "The Human Factors of Computer Graphics Interaction Techniques", IEEE Computer Graphics (CG & A), November 1984, pp. 13-48. (A good overview, including higher level topics which I have postponed to a later article. Excellent bibliography.)

J. D. Foley and A. Van Dam, FUNDAMENTALS OF INTERACTIVE COMPUTER GRAPHICS, Addison Wesley, 1984, Chapters 5 and 6. (If you can't get the article above, read this. If you are designing graphics apps, buy the whole book! Staggering bibliography.)

Ben Schneidermann, "Direct Manipulation: A Step Beyond Programming Languages", IEEE Computer, August 1983, pp. 57-69. (What do Pacman and Visicalc have in common? Schneidermann's analysis is vital to creating hot interfaces.)

THIS WEEK'S QUOTABLE QUOTE

HOFSCHEDTER'S LAW

"VE GET TO SOON OLD... UND TOO LATE SCHMART!"

translated by "sammy's crew"
